Introduction

In many everyday scenarios, Calc spreadsheets can be used to aggregate sets of data and to perform analyses on them. As the data in a spreadsheet is laid out in a table view, plainly visible and easily edited or extended, some users may not need the comprehensive relational database facilities provided by the Base component of LibreOffice. For such users, Calc has sufficient functionality to act as a simple yet capable database-like platform. This chapter presents an overview of these capabilities and explains them using LibreOffice Basic macros and user interface examples.

For those users who initially choose to manage their data in a Calc spreadsheet and subsequently decide that they need to use a more comprehensive database system, migrating Calc data to Base is straightforward. See the *Base Guide* for more information.

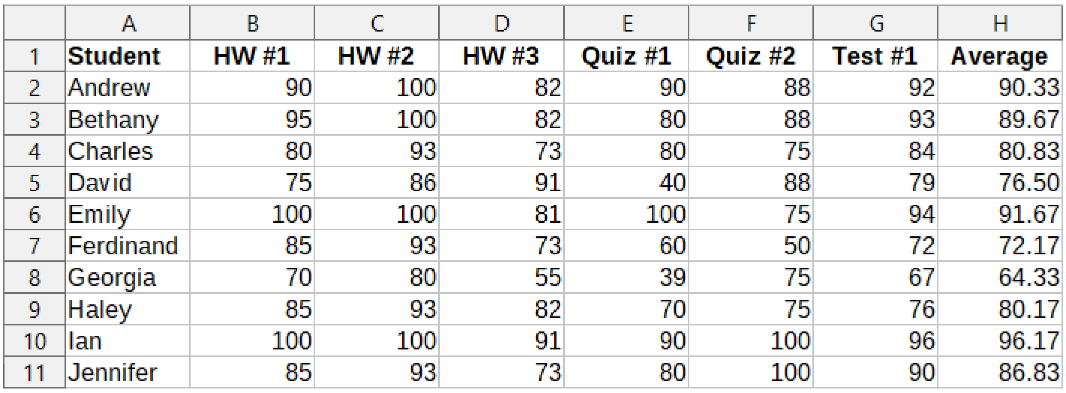
Though it was created for macro programmers, this chapter is intended to be accessible for all users. If you do not want to use macros, simply skip the sections that deal with them. However, if you are interested in learning more about them, see Chapter 12, Macros, and Andrew Pitonyak's book, *OpenOffice.org Macros Explained* (OOME). Much of the macro information in this chapter is drawn or adapted from the OOME and LibreOffice's API reference at https://api.libreoffice.org/docs/idl/ref/index.html.

A database primer

In a typical database, related data is organized into tables, which are arranged in a grid-like series of rows and columns similar to a spreadsheet. Each row of a table represents a data record, while each column represents a field within each record. Each cell in a field contains an individual data item or attribute, such as a name, while each record consists of related attributes that correspond to a single entity, like a person. A database table tends to have a fixed number of fields, but can have an indefinite number of records.

While a table may have hundreds or thousands of rows, individual records can be easily found, retrieved, and updated using information requests, called queries, that search for records that meet a specified set of criteria. It is this ease of access that makes a database table more useful than simply filing away information in an unordered spreadsheet.

To illustrate this concept of a database table, consider the example of a class grading sheet (Figure 452). In this sheet, each row represents individual students taking the class, while each column contains their names and grades. With this table, you can quickly look up individual students' grades simply by searching for their names, and you can determine which students are passing the class by filtering out records with failing average scores.



	A	B	C	D	E	F	G	H
1	**Student**	**HW #1**	**HW #2**	**HW #3**	**Quiz #1**	**Quiz #2**	**Test #1**	**Average**
2	Andrew	90	100	82	90	88	92	90.33
3	Bethany	95	100	82	80	88	93	89.67
4	Charles	80	93	73	80	75	84	80.83
5	David	75	86	91	40	88	79	76.50
6	Emily	100	100	81	100	75	94	91.67
7	Ferdinand	85	93	73	60	50	72	72.17
8	Georgia	70	80	55	39	75	67	64.33
9	Haley	85	93	82	70	75	76	80.17
10	Ian	100	100	91	90	100	96	96.17
11	Jennifer	85	93	73	80	100	90	86.83

Figure 452: Grading sheet example